



COMPUTER CENTRE

BULLETIN

Vol. 6, No 6
12 June 1973

Editor:
Sarah Barry

SYSTEM PERFORMANCE

A request has been received that we define the terms used in the system performance report.

ON TIME - The total time that the system is powered up for maintenance or use.

GOOD TIME - ON TIME less LOST TIME through scheduled and unscheduled maintenance or systems crashes.

USEFUL TIME - GOOD TIME less OPERATIONAL OVERHEADS.

TIME BETWEEN CRASHES - GOOD TIME/NUMBER OF CRASHES.

It should be noted that more detailed figures are maintained within the Centre for management purposes. A review of the information given in the Newsletters and Bulletins on performance is not really indicative of the availability of the system from the user viewpoint. For example, all idle time on the equipment is used for running engineering diagnostics and in the past this has been included as scheduled maintenance.

In a time-shared system the most important indicators of performance are the number of system failures, the mean time between crashes and the mean time to recover from failure.

Of course, such figures must be interpreted against a standard. In a normal month of 480 hours of ON TIME, we would expect 48 hours of scheduled or unscheduled maintenance and 40 hours of operational overhead (including accounting). 20 crashes over a period of 432 hours operation would not be unreasonable based on performance of similar equipment throughout the world. A mean recovery time of 10 minutes is reasonable under present conditions.

Therefore, we would expect that for a normal month -

USEFUL TIME	392 HRS	82%
GOOD TIME	432 HRS	90%
TOTAL ON TIME	480 HRS	
TOTAL CRASHES	20	
MEAN TIME BETWEEN CRASHES	21.6 HRS	
MEAN TIME TO RECOVER	10 MINS	

Unfortunately, we have reached the conclusion that in systems such as ours, there is no such thing as a normal period. Particularly, is this so in the situation we have of changing to a new operating system, implementing new services and installing additional hardware.

The performance figures for May were as follows:

USEFUL TIME	404 HRS	80%
*GOOD TIME	458 HRS	91%
TOTAL ON TIME	503 HRS	
TOTAL CRASHES	16	
MEAN TIME BETWEEN CRASHES	29 HRS	
MEAN TIME TO RECOVER	20 MINS	

*Adjusted for idle time when engineering diagnostics are running.

The Centre is of course very happy with this performance. In terms of systems reliability figures usually quoted (i.e. ignoring scheduled maintenance), we achieved a 99% reliability figure for the full month. The mean time to recover is higher than desired but we expect this to reduce significantly when 5-series operating system is introduced. We also expect that 5-series and the new accounting system will allow considerable reductions in operating overheads.

EXPONENTIATION IN FORTRAN

[WN-135]

Where the exponent is a constant, the compiler realizes the exponentiation as a series of in-line multiplications. If, however, the exponent is a variable, a call to a system routine is generated. This routine, if the exponent is zero, returns the value of one without further checking.

Thus the statement

B=A**I

will set B to 1.0 for I=0 - even if A is also equal to 0.0.

COBOL VERSION 5

[WN-140]

Cobol Version 5 is available on the NEW directory. The standard Cobol operating system (Lib015) has been patched to allow better handling of magnetic tapes containing multiple files when the 'I-O-CONTROL - MULTIPLE FILE TAPE' clause is not used. During testing of the new version, a Cobol program using a sort was observed to run incorrectly. Reports have been received that occasionally incorrect results have been occurring with Cobol programs using a sort. Would any user who suspects that he has had a problem of this nature please report it, so that we may attempt to isolate the problem.

STATISTICAL PACKAGE

[WN-135]

The statistical package for the social sciences has been released as category 4. Note that it uses more than 29K of core, thus it can only be run after 6 p.m. It has been temporarily released onto the system directory, but will later be put onto the STATS directory. All necessary documentation for using the package is in SPSS, statistical package for the social sciences by Nie, Bent and Hull published by McGraw-Hill.

SSP MANUAL

[WN-140]

The SSP manual has again been 'removed' from the clients' room. It will not be replaced - would the person who 'removed' it kindly return it as soon as possible.

1 CARD READER

The existing card reader (SOROBAN) has always been a difficult unit to maintain and very frustrating to operate. We have redesigned the vacuum system and replaced the complete card transport mechanism during its four years of operations, in an attempt to keep the reader going. The stage has been reached where replacement is essential.

After review of the characteristics and price of a number of readers the DOCUMENTATION model M1000C reader was considered the most suitable. A second reader capable of reading cards with holes or pencil marks has also been ordered to cater for the MIDITRAN type of operation. This is a DOCUMENTATION model IOM1000C. Delivery will be in September and both are expected to be in operation by the end of September.

Briefly, both readers are console types, operating at 1000 cards per minute and have a stacker/hopper capacity of 1500 cards. Cards can be loaded and unloaded while running. One useful feature which operators will appreciate is that the card reader will attempt 6 pick operations automatically on the one card before giving a pick failure. An infra-red light source is used in the read station, the vacuum and transport mechanism are very simple, both resulting in high reliability.

One point about the optical mark reader should be clarified. Only specially prepared mark sense cards containing punched holes or marks may be read. The major consideration in the mark sense card specification is the reflectivity of the card itself, and all ink must be reflective. On normal cards as used in the Centre, the ink is non-reflective.

The optical mark reader should also provide a needed service in collection of survey data and multiple choice examinations. It should also be noted that it is still proposed to install a mark page reader in 1975.

2 PLOTTER

To enable the Centre to provide better quality plotting and a larger plot area, a CALCOMP 936 drum plotter has been ordered. Delivery will be late September and operational in November.

The width of the plotting area is 33 inches, compared with 11 inches on the existing plotter. The step size is 0.05 mm which is approximately 1/5th of the step size of the existing plotter. Operationally, the paper is held down by vacuum (no sticky tape required) and operator controls allow movement in one of 8 directions by pushing one of eight switches and the speed of the pen, in local modes, can be controlled by a simple sliding control.

B6-6
12Jun73

The plotter also has 3 pens, any one of which can be selected under program control, one at a time. The pens will be used to provide different colours, or different pen thicknesses of the one colour.

3 PORTABLE TERMINAL

A portable terminal has been ordered to cater for demonstrations in departments, schools etc., where a non-permanent connection is required. Data is transmitted over telephone lines, and is coupled via an acoustic coupler in the terminal, through the handset of a telephone. By this method, the terminal may be located wherever there is a telephone, e.g. a private home, or even a public telephone booth.

4 CORE MEMORY

The AMPEX 64K core memory installed in February has completed all acceptance tests successfully and has more than doubled our core memory capacity, which previously consisted of three 16K DEC memories. Nominally rated at 650 nanosecs. full cycle time, the memory timing has been checked at 560ns full cycle, 260 ns read access. When compared with the MA10's at the memory bus, the AMPEX read access is 250 ns (MA10 is 440 ns).

5 MAGNETIC TAPES

Two additional 9 track magnetic tapes will be installed later this year. Investigation into the type to be ordered is now in progress.

If you have any points which you wish to raise regarding our equipment programme, please contact Alan Coulter, Director of the Centre.

EDIT SUGGESTION

[WN-137]

sug: An ADD capacity should be added to the Editor. ADD AFTER (abbreviated to ADA) for adding to the end of a line. ADD BEFORE (abbreviated to ADB) for adding statement numbers. Default option should be ADA, i.e. AD gives ADA. This facility would save time and money where at present it would be necessary to use REPLACE.

rep: This is a worthwhile suggestion and will be considered together with other proposals that have been made. However, the Editor is presently at the limit of the 2K boundary and it is unlikely that this change alone would be made. It is possible to append character strings at the beginning of a line by changing the null string.
viz: in your example of adding a statement number
*C //007 /

TELEPHONE NUMBER CHANGES

[WN-140]

As a preliminary step towards the introduction of in-dialling facilities on the University's PABX, it is necessary for the PMG to change the number of certain extensions.

Computer Centre numbers that have been changed are:

old	person	new
687	Mr E.J. Sokoll (Senior Engineer)	187
688	Office and general enquiries	188
689	Mr A.W. Coulter (Director)	189
	Miss J. Dixon (Secretary)	189
8391	Mr J.D. Noad (Senior Systems Analyst)	6391
8168	Mr C.C. deVoil (Operations Manager)	6168
8471	Mr J.J. Jauncey (Operations Supervisor)	6471
8101	Computer System Status	6101
8288	Programming Staff	6288
	Program Librarian	
8166	Administrative Officer	6166
8472	Maintenance Technicians	6472

STANDARD COMPLEX ALGEBRA TRANSLATER FOR PDP-10

C.J. Andrews*

1 INTRODUCTION

SCAT is a complex arithmetic processor, which should be of use to those involved in repetitive calculation using complex numbers, e.g. electrical engineering circuit calculations. Such calculations are tedious to perform in large quantities. An approach to relieving this tedium would be to program each task for a computer and then run each task

*Year IV Student, Dept. of Electrical Engineering

separately. This means, however, that a separate program has to be developed, compiled and perhaps debugged for each problem. No saving of effort is obtained.

With this in mind SCAT was developed. It can be used for any complex arithmetic operation. A command, consisting of an expression to be evaluated, is input to the program, which responds with the result. The result may be stored for future use, thus allowing long expressions to be evaluated in parts. Certain special commands are available for special tasks.

This documentation should enable a user to run SCAT and obtain benefit from it. Comments regarding the program, or this documentation, are welcomed and may be directed to Project 613 via the PDP-10 mail system.

2 COMMANDS

The general form of a command to SCAT is

<expression><assignment>

The command is in two parts and each of these will be dealt with separately. The expression part gives the operation to be evaluated and is mandatory for all, other than special, commands. The assignment part is optional for all the different types of expression available and causes storage of the result for future use if desired.

2.1 ASSIGNMENT PART

The assignment part follows an expression part and is of the form

' = variable-name' ,

where the ' = ' causes the result of the expression to be stored as the variable of the given name.

Variable names are restricted to being one character in length and only the alphabetic characters A-Z may be used. The ' = ' must be delimited by one or more blanks, tabs, or one comma.

2.2 EXPRESSION PART

The expression tells SCAT which arithmetic operation is to be performed and on which values. This yields a result, which is output and/or stored (see 2.1).

The values to be operated on may be

- (a) complex constants (2.2.1)
- (b) previously assigned values (2.2.3)
- (c) a mixture of the two (a) and (b) (2.2.2)
- (d) simple assignment is also allowed (2.2.4)

2.2.1 TWO CONSTANTS

The general form of expression for operation between two constants is:

$(\nabla W.W, X.X\nabla) \nabla O \nabla (\nabla Y.Y, Z.Z\nabla)$

W, X, Y, Z are normal floating point numbers, O is an operator and ∇ represents a mandatory blank character. This blank may also be a tab, or comma (etc).

Valid operators are ALGOL-like, i.e. $:+$, $:-$, $:\ast$, $:/$. The operator must be separated from the constants by blanks, etc.

2.2.2 USE OF VARIABLES

Either constant may be replaced by a variable which has been previously defined, e.g.

(i) $A \nabla O \nabla (\nabla X.X, Y.Y\nabla)$

(ii) $(\nabla W.W, Z.Z\nabla) \nabla O \nabla A$

The name of the variable is typed instead of one of the constants, delimited by blanks (∇).

2.2.3 TWO VARIABLES

Both of the variables may be replaced by variables with delimiting blanks

$A \nabla O \nabla B$

2.2.4 SIMPLE ASSIGNMENT

Values may be assigned to variables in one of two ways, depending on what type of value is to be defined (e.g. pre-defined or constant).

(i) $A \nabla = \nabla B$

assigns the current value of A to B

(ii) $(\nabla W.W, Z.Z\nabla) \nabla = \nabla A$

assigns $W.W+jZ.Z$ to A

If the assignment part of these commands is omitted, the current value of the variable is returned or the value of the constant is printed.

2.3 SPECIAL COMMANDS

2.3.1 DUMP

If 'DUMP' is typed as a command, SCAT will type the current value of all defined variables in the following form:

```
*DUMP
  var-nam-1      real-part-1      imag-part-1
  var-nam-2      real-part-2      imag-part-2
*
```

2.3.2 EXIT

When it is desired to return to monitor from SCAT type 'EXIT'.

3 PROGRAM OUTPUT

3.1 RESULTS OF CALCULATION

SCAT returns the result of a desired calculation in one of two ways, depending on whether assignment has been made or not.

3.1.1 NO ASSIGNMENT

If no assignment has been made, SCAT replies:

```
RESULT ( X.X) + ( Y.Y)J
```

3.1.2 ASSIGNMENT

If an assignment (e.g. to variable C) has been made, SCAT replies:

```
C = ( X.X) + ( Y.Y)J
```

3.2 ERROR MESSAGES

The following messages are output as diagnostics and are self-explanatory:

ILLEGAL ASSIGNMENT

PARENTHESES ENCLOSE TWO CONSTANTS

ILLEGAL OPERATOR

ILLEGAL VARIABLE

VARIABLE NOT PREVIOUSLY DEFINED

4 PROGRAM EXAMPLE

.613. SCAT

SCAT (V2,M0) IS READY

* (2.,3.) :+ (7.,9.) = A

A = (9.) + (12.)J

* A :- (6.,11.) = Q

Q = (3.0) + (1.0)J

*DUMP

<u>A</u>	<u>9.00</u>	<u>12.00</u>
<u>Q</u>	<u>3.0</u>	<u>1.0</u>

*EXIT

EXECUTION TIME .17 SECS

TOTAL ELAPSED TIME 1 MINS 32 SECS

NO EXECUTION ERRORS DETECTED

EXIT

↑C

⋮

5 ACKNOWLEDGEMENTS

Grateful acknowledgement is made to the following people:

- (i) Dr C.G. Wilson - for constant constructive criticism in writing the program and preparing documentation.
- (ii) Mr R.A. Barham - for use of 'FRED'.
- (iii) Mr A.J. Mortimer - for suggestions regarding the syntax of SCAT.

